

# Mustererkennung - Aufgabenblatt 11

André Hacker und Dimitri Schachmann

## 1. Merkmale für Rauch

In der Aufgabenstellung gibt es viele Unbekannte: Wie sehen Labels aus, wie sollen wir klassifizieren, aus welchen und wie vielen Daten sollen wir lernen, bleibt es bei 3 Bildern pro Sequenz, usw. Einige Fragen wirken sich auch auf die Wahl und Modellierung der Features aus. Wir beschreiben und modellieren daher zuerst das Problem, und treffen entsprechende Annahmen.

Bei jedem Machine-Learning Problem müssen folgende Fragen zuerst individuell geklärt werden:

- **Problemtyp:** Wir modellieren es als Multiclass classification problem. Jeder Pixel wird einer Klasse 'Rauch' oder 'Kein Rauch' zugeordnet (Siehe "Gestalt der Labels"). Bei  $d$  Pixeln gibt das  $2^d$  mögliche Ausgaben, die man als Klassen interpretieren kann. Ein übergeordneter Classifier kann dann auf Basis eines Schwellenwertes entscheiden ob das Bild insgesamt als Rauch klassifiziert wird oder nicht. Man kann die Ausgaben reduzieren, indem man die Ausgabe und die Label verkleinert, d.h. man sagt dann nur aus ob in einem bestimmten Bereich es Ursprungsbildes Rauch ist oder nicht.
- **Gestalt der Eingabedaten** (Vor der Vorverarbeitung): 3 Bilder in Sequenz, Graustufen, mit konstanter Bildgröße  $w \times h$ . Kann alles in einen Vektor serialisiert werden.
- **Gestalt der Labels:** Bisher gibt es keine Labels. Unser Vorschlag: Eine Matrix in der Größe der Bilder ( $w \times h$ ) die jeden Pixel klassifiziert mit Werten 0 (kein Rauch) und 1 (Rauch). Die Matrix kann als Vektor serialisiert werden.
- **Anzahl verfügbarer Daten:** Aktuell sind nur 3 Bildsequenzen vorhanden, was für das Training mit den uns bekannten Klassifikatoren völlig inakzeptabel ist. Damit kann kein Klassifikator trainiert werden (Verweis auf VC-Theory, die u.A. besagt, dass ein mächtigeres Modell mehr Eingabedaten braucht um den Generalisierungsfehler ggü. einem einfacheren Modell nicht zu erhöhen. Die Modelle die wir kennengelernt haben brauchen wesentlich mehr Daten für akzeptable Generalisierungsfehler.). Eigentlich ist es auch viel zu wenig um sich Gedanken über gute Features zu machen. Annahme: Wir bekommen noch mehr Daten.
- **Fehlermaß:** Wie sieht das gewünschte Fehlermaß aus (spielt eine Rolle für den Klassifikator, der ja den Fehler minimieren soll): False negatives sollten sehr stark bewertet werden um das Feuerrisiko zu minimieren. False positives hingegen sind akzeptabel, da ein Fehlalarm schnell durch einen Blick auf die Überwachungskamera identifiziert werden kann und niemand dort hinfahren muss.

Im folgenden verweisen wir immer wieder auf die Beispielbilder, die weiter unten angehängt sind. Die Bilder laufen zuerst durch eine **Vorverarbeitung**:

- **Verwacklung erkennen und entfernen:** In Bildsequenz 2 ist die Kamera stark verwackelt, was zu großen Differenzen führt (z.B. sieht man die Umrisse des Hauses). Aber auch in Bild 1 erkennt man dass der Weg verschoben ist. Unser Vorschlag: Man verschiebt eines der Bilder ein paar Pixel in verschiedene Richtungen (evtl. mit Drehung), berechnet jeweils die Differenz zum anderen Bild, und prüft ob sich die Summe aller Differenzen signifikant vermindert (Prozentualer Schwellenwert).

- **Farbtiefe reduzieren und Wertebereich ausnutzen:** Der Bereich der möglichen Zahlen wurde nicht ausgenutzt, so dass wir normalisiert haben um das volle Spektrum auszunutzen. Und wir haben die Farbtiefe auf 8bit Graustufen reduziert. Diese Vorverarbeitung haben wir implementiert (Siehe Funktion `normalize(X, bits)`).
- **Rauschen entfernen:** Hierfür kann ein Gauß Filter angewendet werden. Haben wir nicht implementiert.

Nun bilden wir die **Differenzen** für eine Bildsequenz von 3 Bildern. Dies haben wir implementiert (siehe Funktion `compare` im Anhang). Man sieht, dass der Rauch auf den Differenzbildern hervorragend zum Vorschein kommt, aber eben auch vieles Anderes. Die Differenzen sind die Grundlage für die Feature-Extraction:

- Wir berechnen Differenzen von Bild 1 zu Bild 2 ('Differenzbild1'), von Bild 2 zu Bild 3 (Differenzbild2), und von Differenzbild 1 zu Differenzbild 2 (im folgenden '**summiertes Differenzbild**'). Die Differenz von Bild 1 zu Bild 3 haben wir ignoriert (kann man auch noch hinzunehmen).
- Bei jeder Differenz können negative Werte resultieren. Wir nehmen den absoluten Wert, um Differenzen in beide Richtungen zu unterscheiden. Aktuell unterscheiden wir noch nicht in welche Richtung die Differenz geht (kann man auch machen).
- Wir filtern mit der Funktion `noiseThreshold` kleine Differenzen heraus. Das könnte man verbessern indem man die Verteilung der Intensität ansieht und dann dynamisch den Threshold ermittelt. Sicher gibt es hier auch bessere Filter (Gaussian?) um das Rauschen zu entfernen.

Wir schlagen vor für das Differenzbild der Differenzbilder die **Zusammenhangskomponenten** (connected-regions) zu berechnen. Bei allen Sequenzen sieht man, dass Rauch immer eine connected-Region darstellt. Allerdings braucht man einen geeigneten Algorithmus mit Threshold, damit der Rauch nicht auf zu viele Regionen aufgeteilt wird.

Jede Region hat dann eine Nummer und eine Größe (Anzahl der dazugehörigen Pixel. Nach dieser Vorverarbeitung kommen wir zu den eigentlichen Features:

**Finale Features:** Gegeben eine Sequenz von 3 Bildern ( $w \times h$ ), berechnen wir für jeden Pixel einen Vektor mit Zahlen. Die resultierenden  $w \cdot h$  Vektoren sind die Eingabe für den Klassifikator. Wir schlagen folgende Features **pro Pixel** vor:

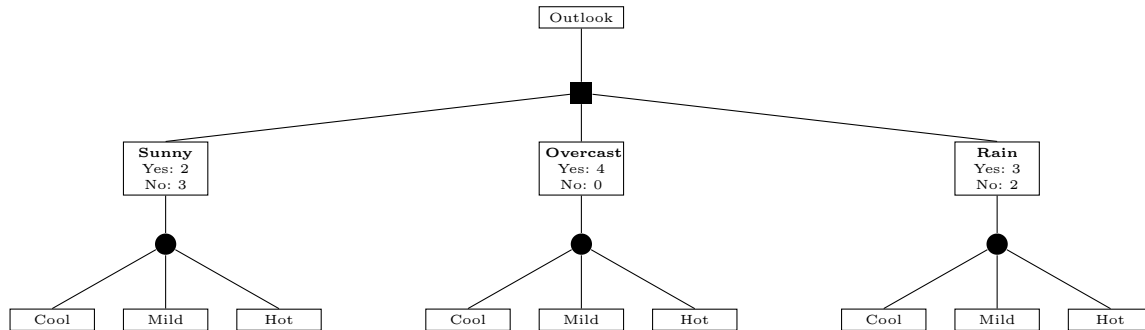
- **Höhe der Differenz** (nach Vorverarbeitung) vom summierten Differenzbild. Evtl. zusätzlich von Bild 1 zu Bild 2 und von Bild 2 zu Bild 3.
- **Größe der Zusammenhangskomponente** im summierten Differenzbild in der sich der Pixel befindet. Über dieses Feature kann der Klassifikator sehr kleine Zusammenhangskomponenten herausfiltern.
- **Wachstum der Zusammenhangskomponente:** Für jede Zusammenhangskomponente prüfen wir, wie sich ihre Größe von Differenzbild1 zu Differenzbild2 vergrößert hat (Zuordnung nicht trivial). Der Faktor ist unser Feature. Hierfür wären mehr Bilder pro Sequenz von Vorteil...
- **Farbwert:** Median des Grautons aller Pixel der Zusammenhangskomponente in der sich der Pixel befindet. Offen bleibt hier, welches Ursprungsbild man betrachtet für die Farbe. Das Feature ist sinnvoll, weil bestimmte Grauwerte bei Rauch sehr häufig sind und andere vermutlich selten. Grundsätzlich wird Rauch eher nicht schwarz sein (ermöglicht Abgrenzung von Schatten)
- **Messwert für den Detailgrad an Struktur:** Wir denken, dass Rauch den Hintergrund verwischen lässt, also die Entropie/Unruhe geringer ist. Sonne hingegen lässt Struktur nur noch deutlicher werden, sollte also eine höhere Unruhe haben. Hier muss also die Textur analysiert werden.

Wir hatten noch mehr Ideen in Richtung Verhältnis der Textur einer Zusammenhangskomponente zu der ihres Umfeldes, Erkennung von sich bewegenden Bereichen, usw., haben uns aber keine einfache Beschreibung ausdenken können.

## 2. ID3

Da es in diesem Anwendungsbeispiel nur zwei Merkmale gibt, muss erstmal nur festgestellt werden, welche der beiden Merkmale die Wurzel des Baumes bildet. Der Rest ergibt sich von selbst. Dafür berechnen wir die Entropie für beide Varianten.

### Outlook als Wurzel



Die verbleibende Entropie berechnet sich wie folgt:

$$E = t_s E_s + t_o E_o + t_r E_r$$

Wobei

$$t_s = \frac{5}{14} \quad t_o = \frac{4}{14} \quad t_r = \frac{5}{14}$$

und

$$E_s = - \left( \frac{2}{5} \cdot \log_2 \left( \frac{2}{5} \right) + \frac{3}{5} \cdot \log_2 \left( \frac{3}{5} \right) \right) \approx 0.971$$

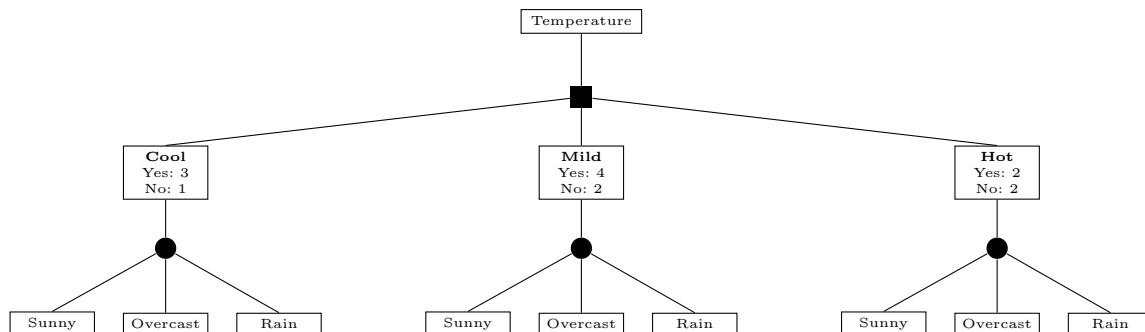
$$E_o = 0$$

$$E_r = - \left( \frac{3}{5} \cdot \log_2 \left( \frac{3}{5} \right) + \frac{2}{5} \cdot \log_2 \left( \frac{2}{5} \right) \right) \approx 0.971$$

Damit:

$$E = \frac{5}{14} \cdot 0.971 + 0 + \frac{5}{14} \cdot 0.971 \approx 0.694$$

### Temperature als Wurzel



Die verbleibende Entropie berechnet sich wie folgt:

$$E = t_c E_c + t_m E_m + t_h E_h$$

Wobei

$$t_c = \frac{4}{14} \quad t_m = \frac{6}{14} \quad t_h = \frac{4}{14}$$

und

$$E_c = - \left( \frac{3}{4} \cdot \log_2 \left( \frac{3}{4} \right) + \frac{1}{4} \cdot \log_2 \left( \frac{1}{4} \right) \right) \approx 0.811$$

$$E_m = - \left( \frac{4}{6} \cdot \log_2 \left( \frac{4}{6} \right) + \frac{2}{6} \cdot \log_2 \left( \frac{2}{6} \right) \right) \approx 0.918$$

$$E_h = 1$$

Damit:

$$E = \frac{4}{14} \cdot 0.811 + \frac{6}{14} \cdot 0.918 + \frac{4}{14} \cdot 1 \approx 1.054$$

### Der Rest des Baumes

Da wir nun festgestellt haben, dass nach dem ID3 Algorithmus *Outlook* die Wurzel des Baumes bildet, müssen wir noch den Rest des Baumes vervollständigen:

